A Heat Transfer Algorithm with a Least Squares Temperature Gradient Calculation for Particle-based Numerical Simulations

Matthew S. Mason[*1], Kuan Chen[2], Patrick G. Hu[3], Liping Xue[4], QingDing Li[5]

[*1,2]Department of Mechanical Engineering, [*1,2]University of Utah, [3,4,5]Advanced Dynamics Incorporated
Salt Lake City, Utah, 84112, USA, Lexington, Kentucky, 40511, USA

[*1]matthew.mason@utah.edu; [2]chen@mech.utah.edu; [3]patrick.g.hu@advanceddynamics-usa.com

Abstract

Fluid-structure-interaction (FSI) is a very important consideration in the modeling of fluid flow over and through contemporary aircraft and aerospace vehicles. The numerical code used in this research utilizes a Lagrangian particle-based solver in conjunction with an Eulerian mesh-based solver (known as the Material Point Method (MPM)) to gain the advantages that both solvers offer to reduce computational overhead. Heat transfer between a fluid and a solid is an important FSI consideration, especially for high velocity (supersonic and hypersonic) aircraft and aerospace vehicles. A least squares gradient calculating technique was utilized combined with Fourier's Law of Conduction to incorporate heat transfer into the MPM algorithm. Application of the heat transfer algorithm to hypersonic single ramp and double cone flows was investigated. The MPM results were compared to experimental and numerical data from independent sources and observed trends agreed well with the results from these sources.

Keywords

Fluid-structure-interaction; Material Point Method; Heat Transfer; Hypersonic Double Cone

Nomenclature

$a_0, a_1, a_2, a_3, a_4, a_5$ = the coefficients to be solved for in the least squares equations

$\mathbf{a}$ = acceleration vector

$\mathbf{b}$ = body forces vector

D = particle diameter for Knudsen number calculations

e = internal energy of material point

e_{res} = the residual error of the assumed solution fit

ε = strain

$\dot{\varepsilon}$ = strain rate

$\mathbf{f}^{ext}$ = external force vector

$\mathbf{f}^{\text{int}}$ = internal force vector

F = the evaluated value for the assumed least squares solution fit

G_{gp} = gradient of the shape function

γ = ratio of specific heats

$\mathbf{I}$ = second order unit tensor

Kn = Knudsen number

k = thermal conductivity

k_b = Boltzmann constant

L = characteristic length

m = mass

μ = absolute viscosity

ρ = density of material point

$\mathbf{p}$ = momentum

p = pressure

$q^{''}$ = heat flux

σ = Cauchy stress tensor

S_{gp} = shape function for transferring values between material points and grid

S_r = sum of squares of the residuals

T = temperature of material point

T_s = temperature at solid surface (wall temperature)

T_∞ = temperature of ambient flow

t = time variable

u_c = calculated solution value

u_e = experimental solution value

$\mathbf{v}$ = velocity vector

$\mathbf{x}$ = position vector

x = x-position variable used in least squares algorithm

y = y-position variable used in least squares algorithm

Introduction

The Material Point Method (MPM) has drawn much interest in recent years for modeling solid and fluid dynamics problems due to the innate ability to monitor individual particle positions without loss of accuracy or difficulty in coding. Within MPM, the momentum equation is solved on an Eulerian mesh; velocity and position information is transferred back to the individual particles. The MPM code used in this research (referred to hereafter as the Aeroelastic Material Point Method (AEMPM)) is uniquely able to perform these calculations for both solid and fluid phases within a single numerical algorithm, which allows fluid-structure interaction in the momentum (force) equation. However, AEMPM does not currently include capability to transfer heat (energy) between the solid and fluid. The motivation for the current research is the development of an integral heat transfer algorithm within a fluid and solid particle based numerical solver. This would allow both momentum and heat transfer phenomena to be solved in a single numerical code without the need for numerical coupling between multiple solvers.

This paper discusses a method to incorporate heat transfer into a MPM code and the results generated. A least squares gradient technique is used to calculate temperature gradients and a Fourier's Law of Conduction heat transfer algorithm utilizes this least squares result to calculate heat transfer between the fluid and solid. Aerodynamic heating by viscous shear and radiation heat transfer are not modeled. The hypersonic flows considered in this paper are laminar and so turbulence effects are not modeled. Verification studies of the algorithm with a 35° ramp and a 25°/55° double cone in a hypersonic flow of nitrogen are performed with numerical and experimental data.

The MPM code utilized in this research is a Navier-Stokes based continuum solver. This type of code was selected in lieu of a Direct Simulation Monte Carlo (DSMC) particle based code since the flows analyzed are in the continuum regime. Additionally, it is very impractical to use a DSMC solver on denser flows (Knudsen number less than 0.01) due to the large number of particles necessary to resolve the flow physics.

State-of-the-art Review

Numerical modeling of heat transfer is significant within a code designed for hypersonic vehicle analysis. Hollis et al. found the peak heating load for hypersonic flight of the X-33 aerospace vehicle to be around 100,000 W/m^2. With such a large heat load on the vehicle, accurately modeling the heating effects during flight to determine material capabilities and thermo-mechanical fatigue effects is vital.

In order to integrate heat transfer into a computational architecture, temperature values or gradients must be known. A least squares algorithm was integrated during the preliminary development of the AEMPM code (see multiple papers by Hu et al.) to calculate property gradients of velocity, density, and energy based upon techniques discussed in the paper by Koh and Tsai. Koh outlined a least squares curve-fitting technique and further defined the setup and numerical implementation of surface boundary conditions into a Lagrangian algorithm.

During the development of a numerical algorithm, it is necessary to obtain (or generate numerically and experimentally) comparable cases that can be used to verify and validate the algorithm. For hypersonic flows, the single ramp and double cone are commonly used baseline cases in the literature (see Harvey et al., Gnoffo, Candler et al., Nompelis, and Moss et al.). Coleman and Stollery performed a study on a single ramp at hypersonic flow velocities and found heat transfer rates on the order of 500,000 to 1,000,000 W/m^2. Moss et al. studied hypersonic heat transfer of both the single ramp and the double cone. The double cone was very similar in geometry to a single ramp (from a two-dimensional perspective) with the addition of a second ramp downstream and heat transfer rates were of similar orders of magnitude as the single ramp.

Moss et al. utilized a Direct Simulation Monte Carlo (DSMC) for Mach 25 flow over a control surface (i.e., an elevator or aileron). The numerical simulations were for a 35 degree compression ramp at a low density flow condition, the results of which were compared qualitatively with oil flow pictures. The DSMC numerical code, a general 2-dimensional (known as G2) or general 3-dimensional (known as F3) axisymmetric code, is a particle based method in which particles are populated in discrete cells. Molecular collisions were simulated with the variable hard sphere molecular model and energy exchange between kinetic and internal modes was controlled by the Larsen-Borgnakke statistical model. The surface of the single ramp was defined with a specified constant temperature and full thermal accommodation and diffuse reflection were assumed for the fluid-structure interactions. The single ramp results were found to be sensitive to grid resolution, however, these results showed good qualitative agreement with the oil flow pictures.

Moss et al. utilized a revised version of the G2 code, known as the DS2V code, to analyze several double cone cases. The DS2V code models both time accurate unsteady flow and time-averaged steady flow simulations as well as molecular collisions and energy exchange in the same manner as the G2 code. Double cone cases were used for validation due to the availability of large amounts of experimental and numerical data. The double cone cases also exercised a numerical code's compression modeling capability since the double cone geometry introduced shock wave interactions at supersonic flow velocities. The results in Moss et al. found excellent agreement between the DS2V code and other numerical codes, however, an error of up to 15% was noted for heat transfer results when compared with experimental data. There was a single experimental case where the DS2V results were erroneous by up to 30%, but the experimental results were believed to be incorrect.

Chanetz et al. analyzed a 25°/65° double cone in Mach 10 flow conditions using two Navier-Stokes based continuum numerical solvers and two DSMC solvers (including the G2 code). These computations were compared to experimental data obtained in the R5Ch wind tunnel in which the mean free path was equal to $5*10^{-4}$ meters. Chanetz et al. concluded that this mean free path justified the use of the Navier-Stokes equations as the experiments were performed in the continuum regime. The computational results between the Navier-Stokes continuum and DSMC solvers agreed and trended with the experimental data except for a 33% deviation at the maximum heating rate where the shock wave and boundary layer met (computational solvers predicted 33% low). The computational solvers' prediction of the separation location disagreed with experimental data by about 17%.

Harvey reviewed the state-of-the-art for the hollow cylinder/flare body (similar to an engine inlet) and the 25°/55° hypersonic double cone cases for both Navier-Stokes and DSMC solvers. These cases were in the realm of Navier-Stokes solutions due to their relatively high freestream densities, but close enough to the rarefied realm that DSMC solvers could practically be applied, although, the computational demands were extremely high. The DSMC solvers required very fine resolution to resolve the flow and underpredicted separation lengths even with cell adaptation and collision modeling improvements. The Navier-Stokes solvers tended to exhibit spurious perturbations in the flow due to the flow's steep property gradients. Harvey concluded that the flows studied were insensitive to vibrational temperature fluctuations suggesting that the analyzed flows were vibrationally frozen.

Material Point Method Algorithm

The fundamental governing equations in the Material Point Method are the conservation equations for mass, momentum, and energy (equations 1, 2, and 3, respectively; see Wikipedia reference for a brief description of the Material Point Method). In AEMPM, the time derivatives were discretized using the backward Euler method and the spatial discretization was performed in a manner similar to typical Finite Element Methods using shape functions and the weak forms of the momentum and energy equations (see Bardenhagen and Kober).

$$\frac{\partial \rho}{dt} + \nabla \rho \cdot \mathbf{v} = 0 \tag{1}$$

$$\rho \mathbf{a} = \nabla \sigma + \rho \mathbf{b} \tag{2}$$

$$\rho \dot{e} = \sigma : \dot{\varepsilon} \tag{3}$$

The strain rate in equation 3 is defined in equation 4. For the present investigation, the small deformation assumption was valid and the strain rate relation in equation 4 was used in the calculations.

$$\dot{\varepsilon} = \frac{1}{2}\left[(\nabla \mathbf{v}) + (\nabla \mathbf{v})^T\right] \tag{4}$$

These equations were discretized for material points, the grid nodes on the Eulerian mesh, or both. The index convention for the discretization equations followed that in Hu et al. and were similar to York II et al., in which grid variables have a subscript g and material point variables have a subscript p . The remaining equations in this paper were derived from these two references. The momentum equation (equation 2) was discretized on the grid as shown in equation 5.

$$m_g \mathbf{a}_g = \mathbf{f}_g^{\text{int}} + \mathbf{f}_g^{ext} \tag{5}$$

Equation 5 was modified to a form based on the relation between acceleration and momentum in equation 6. Next, updated values of momentum at the next timestep were calculated as shown in equation 7.

$$\frac{\partial \mathbf{p}_g}{\partial t} = m_g \mathbf{a}_g \tag{6}$$

$$\mathbf{p}_g(t + \Delta t) = \mathbf{p}_g(t) + \Delta t\left(\mathbf{f}_g^{\text{int}} + \mathbf{f}_g^{ext}\right) \tag{7}$$

The above discretization equations required values to be located on the grid; however, the solution domain was initialized with material points, so shape functions (see Bardenhagen and Kober for a description of the shape functions used) must be used to transfer information between the grid and material points. Values of mass and momentum were calculated using the shape functions in equations 8 and 9.

$$m_g = \sum_p m_p S_{gp} \tag{8}$$

$$m_g \mathbf{v}_g = \sum_p m_p \mathbf{v}_p S_{gp} \tag{9}$$

The material point velocity and position were then solved using the grid values of acceleration and momentum as shown in equations 10 and 11.

$$\mathbf{v}_p(t + \Delta t) = \mathbf{v}_p(t) + \sum_g S_{gp} \mathbf{a}_g \Delta t \tag{10}$$

$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + \sum_g S_{gp} \frac{\mathbf{p}_g}{m_g} \Delta t + \frac{1}{2} \sum_g S_{gp} \mathbf{a}_g (\Delta t)^2 \tag{11}$$

The material points have values of updated positions and velocities which were used for calculations of the updated strain values (for the solid material) using equation 12. The term summed with the strain value at time t is known as the strain increment. The fluid material point stresses were calculated as shown in equation 13.

$$\varepsilon_p(t + \Delta t) = \varepsilon_p(t) + \frac{\Delta t}{2} \sum_g \left[G_{gp} \mathbf{v}_g(t + \Delta t) + \left(G_{gp} \mathbf{v}_g(t + \Delta t)\right)^T\right] \tag{12}$$

$$\sigma_p = -p_p \mathbf{I} + 2\mu\dot{\varepsilon} - \frac{2}{3}\mu * tr(\dot{\varepsilon})\mathbf{I} \tag{13}$$

Since the external force vector was an input within the AEMPM code, only updated values of the internal force vector must be calculated for use in the momentum equation. The internal force vector was calculated using the material point stresses in equation 14. Then, the fluid material point density was updated for continuity (equation 1) using equation 15. Note that the value inside the trace (the summation of the main diagonal of a matrix) of the denominator is the strain increment for the fluid.

$$\mathbf{f}_g^{\text{int}} = -\sum_p G_{gp} \frac{m_p}{\rho_p} \sigma_p \tag{14}$$

$$\rho_p(t + \Delta t) = \frac{\rho_p}{1 + tr\left(\varepsilon_p(t + \Delta t) - \varepsilon_p(t)\right)} \tag{15}$$

The material point energy was updated using the new values of density and stress in equation 16 and the material point pressure was calculated with updated values of density and energy using the equation of state as shown in equation 17. The material point temperature was calculated by dividing the material point energy by the specific heat of the material for both fluid and solid phases.

$$e_p(t + \Delta t) = e_p(t) + \frac{\sigma_p(t + \Delta t)\left(\varepsilon_p(t + \Delta t) - \varepsilon_p(t)\right)}{\rho_p(t + \Delta t)} \tag{16}$$

$$p_p = (\gamma - 1)\rho_p e_p \tag{17}$$

Heat Transfer Algorithm

This paper focuses on the fluid-structure interface and the transfer of thermal energy across this interface. The computational layout of the AEMPM algorithm used in this research is shown in Fig. 1. The grid remains fixed throughout the entire calculation and was used as a means of momentum transfer. The material points were initiated in the solution domain with mass and flow properties. Figure 1 shows the separate types of material points present in the algorithm. The fluid and solid MP's represent the fluid and solid phases respectively, while the surface MP's represent the fluid-structure interface. These surface MP's, initiated in the algorithm with fluid properties, were used as the exchange "medium" for energy between the fluid and solid MP's. A "cutcell" is defined as a grid cell in which the fluid-structure interface crosses. For the hypersonic single ramp and double cone cases, the Dirichlet boundary condition of a constant wall temperature was enforced to allow comparison with experimental data and other numerical results that enforced this thermal boundary condition. The solid MP's were omitted from the analyses since a constant wall temperature boundary condition yielded no temperature change within the solid material.

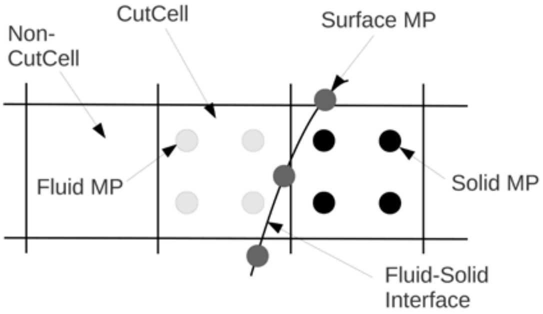


FIG. 1. AEMPM COMPUTATIONAL LAYOUT.

In the present study, as a fluid flows over a ramp and double cone at high speeds, large pressure gradients (strong compression) are developed in the flow which result in strong shock waves, leading to large temperature changes of the fluid. The temperature distribution in the interface region is continuous (i.e., there is no temperature jump). However, the fluid particles in contact with the solid wall may not be the same temperature as the solid because there is a small distance between the fluid and solid particles where a temperature gradient exists. This temperature difference decreased as grid resolution increased and was less than 5% of the wall temperature for the highest grid resolution cases. Viscous shear also generates heat transfer, but this friction heating mechanism is not currently modeled in the numerical algorithm and so translational energy (fluid compression) is the only mode of fluid particle temperature change.

Fourier's Law of Conduction for a Cartesian grid is shown in equation 18 (from Incropera et al.). The heat flux between two points can be calculated using the x- and y- derivatives of the temperature (calculated using the derivatives from the least squares solution in Appendix I), which is shown in equations 19 and 20. The 'a' coefficients in equations 19 and 20 are the same coefficients as those seen in the least squares solution from Appendix I.

$$q^{''} = -k\left(\frac{\partial T}{\partial x} + \frac{\partial T}{\partial y}\right) \tag{18}$$

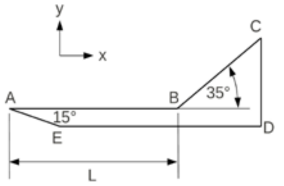
$$\frac{\partial T}{\partial x} = a_1 + 2a_3 x + a_5 y \tag{19}$$

$$\frac{\partial T}{\partial y} = a_2 + 2a_4 y + a_5 x \tag{20}$$

The boundary conditions utilized in the MPM results in this paper were characteristic boundary conditions for the inlet, outlet, top, and bottom fluid domain boundaries (see Anderson Jr. for characteristic boundary condition description). The fluid-structure interface was defined as a Dirichlet boundary condition in which the temperature of the solid surface was fixed.

Application of Heat Transfer to a Single Ramp in Hypersonic Flow

The first verification case was a single ramp at a 35° angle. A single ramp is a simplified model of a hypersonic vehicle's control surface (such as an elevator) and exercises a numerical code's aerodynamic flow and compressibility simulation capabilities. A motivation to simulate these types of flows is driven by the fact that flight safety greatly depends on the effectiveness of control surfaces. Numerical data from Moss et al. has been compared to the results from the AEMPM code in this paper. The 35° single ramp was setup as shown in Fig. 2 with the coordinate locations in meters (L = 0.0714 meters). The flow was nitrogen with constant specific heats and had the following initial conditions: $V_\infty = 1521 \ m/s$, $\rho_\infty = 1.401*10^{-4} \ kg/m^3$, $T_\infty = 9.06 \ K$, $P_\infty = 0.376 \ Pa$, $M_\infty = 24.8$, and $T_S = 403.2 \ K$. Note that the surface of interest is made up of coordinate locations A, B, and C (the heat transfer surface of a flight control surface deflected in the positive y-direction) and the results shown reflected only this surface.



Point	Coordinate
A	(0,0)
B	(0.07140,0)
C	(0.12988,0.04695)
D	(0.12988,-0.006)
E	(0.02239,-0.006)

FIG. 2. SINGLE RAMP CONFIGURATION AND SETUP COORDINATES.

The Knudsen number was calculated using equation 21. For the single ramp flow condition, the Knudsen number was 0.007 and therefore, the continuum assumption (Navier-Stokes equations with the no-slip boundary conditions) is valid (Panton). For reference, Panton classified the flows as follows (Panton, page 680):

$0 \le Kn \le 0.01$ "continuum" flow (Navier-Stokes equations with no-slip boundary conditions)

$0.01 \le Kn \le 0.1$ slip flow (Navier-Stokes equations with slip boundary conditions)

$0.1 \le Kn \le 3$ transitional flow (Navier-Stokes equations are not valid in this regime)

$3 \le Kn \le \infty$ free molecular flow (molecules interact only with walls)

$$Kn = \frac{k_b T_\infty}{\sqrt{2}\pi D^2 pL} \qquad (21)$$

The results of the MPM code and the verification case are shown in Fig. 3 (zoomed in at the ramp transition from x=0.05 m to x=0.13 m). The verification case is another independently developed numerical code (known as the G2 code) which is discussed in more detail in Moss et al. The number following the "MPM Grid=" in the legend is the relative resolution of the grid used in the computational domain. Grid resolution and the number of material points per grid cell are both variables within the AEMPM code. During the studies discussed in this paper, it was found that the optimal material point density was 4 material points per grid cell (2x2 pattern in x- and y-directions). Additional material points per grid cell beyond this caused instability and larger perturbations in the numerical algorithms. The results discussed in this paper utilized four material points per grid cell and varying grid resolutions.

The differences between the G2 code and MPM results are attributable to the differing code formulations. The G2 code, a Direct Simulation Monte Carlo numerical code, includes energy exchange between translational, rotational, and vibrational modes. Moreover, diffuse reflection is assumed for the fluid-structure interface. The AEMPM code only incorporates the translational energy exchange mode and is a Navier-Stokes continuum based numerical solver which excludes diffuse reflection.

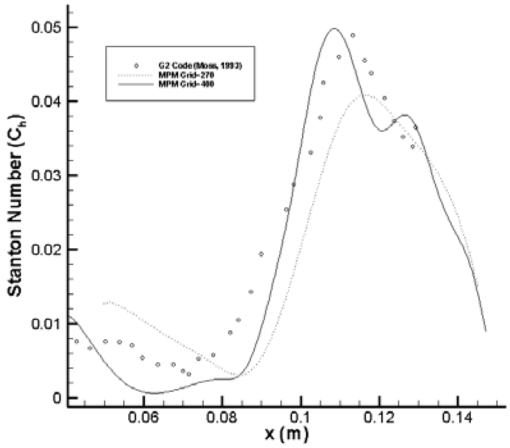


FIG. 3. SINGLE RAMP HEAT TRANSFER DISTRIBUTION.

The L^2 relative error norm was calculated for the MPM data using equation 22 and is shown in Table 1. Increasing the grid resolution by about 50% (from MPM Grid=270 to MPM Grid=400) reduced the relative error norm by about 60%. Note the large reduction in relative error for the peak heat transfer rate (near x=0.12 m in Fig. 3).

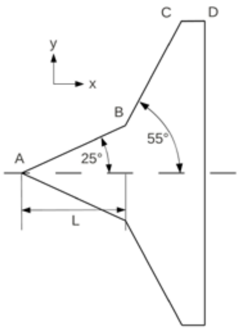
$$L^2 error = \sqrt{\frac{\sum (u_e - u_c)^2}{\sum (u_e)^2}} \qquad (22)$$

TABLE 1. MPM HEAT DISTRIBUTION COMPARISON TO NUMERICAL RESULTS.

	L^2 Relative Error Norm	Relative Error in Peak Heat Transfer Rate
MPM Grid=270	0.265	36%
MPM Grid=400	0.101	4%

Application of Heat Transfer to Double Cone in Hypersonic Flow

The double cone is a popular case for studying hypersonic flows (a few different configurations and cases were discussed in Harvey et al.) as it incorporates complex shock wave interaction between two shocks and the associated heat transfer at the fluid-structure interface. For this investigation, the double cone was setup as shown in Fig. 4 (L = 0.092072 meters). The coordinate locations are in meters and the points below the centerline are the same x- values with a negative y-value. The first cone has a half angle of 25° and the second cone has a half angle of 55°.



Point	Coordinate
A	(0,0)
B	(0.092072,0.042934)
C	(0.153666,0.130900)
D	(0.195000,0.130900)

FIG. 4. DOUBLE CONE CONFIGURATION AND SETUP COORDINATES.

The case studied had the same initial conditions as Moss et al. (nitrogen at $V_\infty = 2072.6 \quad m/s$, $\rho_\infty = 1.757 * 10^{-4} \quad kg/m^3$, $T_\infty = 42.61 \quad K$, $P_\infty = 2.23 \quad Pa$, $M_\infty = 15.6$, and $T_S = 297.2 \quad K$) and constant specific heats. The Knudsen number for the Double Cone flow conditions was 0.004 and so the continuum assumption is valid. The AEMPM results in this paper were compared to the experimental values from Moss et al.'s paper for increasing degrees of grid resolution in Fig. 5.

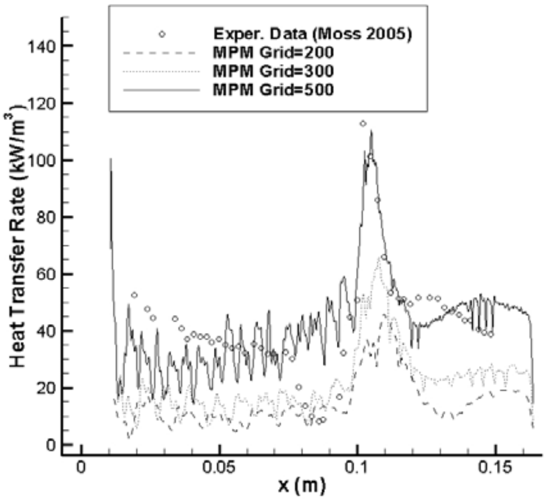


FIG. 5. DOUBLE CONE HEAT TRANSFER DISTRIBUTION.

The results exhibited similar trending as the experimental results from Moss et al.'s paper. The double cone MPM results did not exhibit the "dip" trend in the separation region (the area between x=0.07 m and x=0.1 m) because the AEMPM code does not resolve recirculation or flow separation effects. However, note that the peak heat transfer 'spike' seen at x=0.1 m is resolved within about 6% in the most resolute case. Furthermore, note the oscillations seen in the results which correlate with the observations by Harvey that Navier-Stokes solvers tend to exhibit spurious perturbations due to large property gradients

The L^2 relative error norms and relative errors in peak heat transfer rate calculated for the MPM data are shown in Table 2. Note that increasing the grid resolution from 200 to 500 (approximately six times the MP's in the computational domain) reduced the relative error norm by about 65% and significantly improved the relative error in the peak heat transfer (near x=0.1 m).

TABLE 2. MPM HEAT DISTRIBUTION COMPARISON TO EXPERIMENTAL RESULTS.

	L^2 Relative Error Norm	Relative Error in Peak Heat Transfer Rate
MPM Grid=200	0.584	63%
MPM Grid=300	0.443	48%
MPM Grid=500	0.207	6%

Conclusions

The incorporation of heat transfer within a hypersonic aerospace vehicle numerical model is vital to calculate the effects of the temperature gradients and compression developed within shock waves and pressure gradients at the fluid-structure interface. The least squares gradient method described in this paper was used to calculate the temperature at the fluid-structure interface and heat transfer at this interface. Using a conductive model (Fourier's Law of Conduction) for fluid-structure interface heat transfer resulted in agreeable trends for the single ramp case and a preliminary study in the double cone case.

ACKNOWLEDGMENTS

Funding for this research and the base numerical AEMPM code were provided by Advanced Dynamics Incorporated, Lexington, Kentucky.

Appendix I

Least Squares Gradient Technique

The least squares gradient method is a matrix algebra solution technique used to calculate gradients of a function value. The AEMPM code uses a least squares gradient technique to calculate values of velocity, density, and energy with a Dirichlet boundary condition (known value on the boundary) for velocity and energy as well as a Neumann boundary condition for density. This paper studied the implementation of the least squares gradient technique into the AEMPM code to calculate temperature gradients using the temperature information of neighboring material points. The basic steps used to implement the least squares method for temperature gradients have been outlined below. A second order polynomial was utilized due to its ease of implementation and yielded more accurate results over a first order polynomial. Other functions such as cosine series were investigated, but stability issues were observed and so a second order polynomial was utilized for the implementations discussed in this paper.

Assume a solution of the form:

$$F = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 y^2 + a_5 xy + e_{res} \qquad \text{(AI.1)}$$

Solving this equation for the sum of the squares of the residuals e_{res} by means of the least squares method yields (Chapra and Canale):

$$S_r = \sum_{i=1}^{n} (F_i - a_0 - a_1 x_i - a_2 y_i - a_3 x_i^2 - a_4 y_i^2 - a_5 x_i y_i)^2 \qquad \text{(AI.2)}$$

To solve for the unknowns, the derivatives of this equation are taken with respect to each unknown coefficient yielding the following set of equations:

$$\frac{\partial S_r}{\partial a_0} = -2\sum_{i=1}^{n} (F_i - a_0 - a_1 x_i - a_2 y_i - a_3 x_i^2 - a_4 y_i^2 - a_5 x_i y_i)$$

$$\frac{\partial S_r}{\partial a_1} = -2\sum_{i=1}^{n} x_i (F_i - a_0 - a_1 x_i - a_2 y_i - a_3 x_i^2 - a_4 y_i^2 - a_5 x_i y_i)$$

$$\frac{\partial S_r}{\partial a_2} = -2\sum_{i=1}^{n} y_i (F_i - a_0 - a_1 x_i - a_2 y_i - a_3 x_i^2 - a_4 y_i^2 - a_5 x_i y_i)$$

$$\text{(AI.3)}$$

$$\frac{\partial S_r}{\partial a_3} = -2\sum_{i=1}^{n} x_i^2 (F_i - a_0 - a_1 x_i - a_2 y_i - a_3 x_i^2 - a_4 y_i^2 - a_5 x_i y_i)$$

$$\frac{\partial S_r}{\partial a_4} = -2\sum_{i=1}^{n} y_i^2 (F_i - a_0 - a_1 x_i - a_2 y_i - a_3 x_i^2 - a_4 y_i^2 - a_5 x_i y_i)$$

$$\frac{\partial S_r}{\partial a_5} = -2\sum_{i=1}^{n} x_i y_i (F_i - a_0 - a_1 x_i - a_2 y_i - a_3 x_i^2 - a_4 y_i^2 - a_5 x_i y_i)$$

Finally, setting these equations equal to zero to find the minimum of the sum of the squares and re-arranging them yields the following system of equations (where all summations are from i=1 to n, with n being the number of material points included in the solution):

$$(n)a_0 + \left(\sum x_i\right)a_1 + \left(\sum y_i\right)a_2 + \left(\sum x_i^2\right)a_3 + \left(\sum y_i^2\right)a_4 + \left(\sum x_i y_i\right)a_5 = \sum F_i$$

$$\left(\sum x_i\right)a_0 + \left(\sum x_i^2\right)a_1 + \left(\sum x_i y_i\right)a_2 + \left(\sum x_i^3\right)a_3 + \left(\sum x_i y_i^2\right)a_4 + \left(\sum x_i^2 y_i\right)a_5 = \sum x_i F_i$$

$$\left(\sum y_i\right)a_0 + \left(\sum x_i y_i\right)a_1 + \left(\sum y_i^2\right)a_2 + \left(\sum x_i^2 y_i\right)a_3 + \left(\sum y_i^3\right)a_4 + \left(\sum x_i y_i^2\right)a_5 = \sum y_i F_i$$

$$\text{(AI.4)}$$

$$\left(\sum x_i^2\right)a_0 + \left(\sum x_i^3\right)a_1 + \left(\sum x_i^2 y_i\right)a_2 + \left(\sum x_i^4\right)a_3 + \left(\sum x_i^2 y_i^2\right)a_4 + \left(\sum x_i^3 y_i\right)a_5 = \sum x_i^2 F_i$$

$$\left(\sum y_i^2\right)a_0 + \left(\sum x_i y_i^2\right)a_1 + \left(\sum y_i^3\right)a_2 + \left(\sum x_i^2 y_i^2\right)a_3 + \left(\sum y_i^4\right)a_4 + \left(\sum x_i y_i^3\right)a_5 = \sum y_i^2 F_i$$

$$\left(\sum x_i y_i\right)a_0 + \left(\sum x_i^2 y_i\right)a_1 + \left(\sum x_i y_i^2\right)a_2 + \left(\sum x_i^3 y_i\right)a_3 + \left(\sum x_i y_i^3\right)a_4 + \left(\sum x_i^2 y_i^2\right)a_5 = \sum x_i y_i F_i$$

Note that this system of equations is a symmetric matrix. Therefore, in the solution algorithm only a vector of the upper diagonal matrix needs to be generated. This vector can be generated in a loop very quickly by adding each material point's respective value for the summation in the upper diagonal being solved for (i.e., $\left(\sum x_i\right)$ or $\left(\sum y_i^3\right)$). The right hand side of equation AI.4 is solved in the same manner with summations in a loop. Finally, Cholesky Decomposition (see equations AI.5 and AI.6) is used to solve the system of equations and the original value of F from equation AI.1 above can be obtained. This F is the least squares curve fitted value for a variable at the material point (i.e., temperature in the case discussed in this paper).

Cholesky Decomposition

Cholesky decomposition is a technique for solving symmetric matrices $\left([A] = [A]^T\right)$ which can be decomposed as $[A] = [L][L]^T$. To obtain the individual elements within the $[L]$ matrix, equations

AI.5 and AI.6 are used (where equation AI.5 is for i indices from 1 to $k-1$) from Chapra and Canale.

$$l_{ki} = \frac{a_{ki} - \sum_{j=1}^{i-1} l_{ij} l_{kj}}{l_{ii}} \qquad \text{(AI.5)}$$

$$l_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} l_{kj}^{2}} \qquad \text{(AI.6)}$$

Appendix II

AEMPM Heat Transfer Algorithm Pseudo Code

The integration of the Fourier's Law of Conduction heat transfer algorithm for the fluid-structure interface into the AEMPM code followed the subroutine steps seen in the pseudo-code below:

```
If cell is a cutcell
    For each MP in cutcell
    Loop through a 10x10 grid stencil centered around cutcell
    (loop includes all 100 grid cells in this step)
        Calculate relative position of fluid MP's and surface MP's in cutcell
            Calculate least squares coeff. with surface and fluid MP's
    Solve for unknowns using Cholesky Decomposition
    Calculate heat flux at surface MP using equation 18
```

REFERENCES

Anderson Jr., J., Modern Compressible Flow with Historical Perspective, 1st ed., 1982, McGraw-Hill, pp. 263-290.

Bardenhagen, S., Kober, E., "The Generalized Interpolation Material Point Method," CMES, vol. 1, no. 1, 2000, pp. 11-29.

Candler, G., Nompelis, I., Druguet, M., Holden, M., Wadhams, T., Boyd, I., Wang, W., "CFD Validation for Hypersonic Flight: Hypersonic Double-Cone Flow Simulations," AIAA Journal, AIAA 2002-0581, 2002.

Chanetz, B., Pot, T., Benay, R., Moss, J., "New Test Cases in Low Density Hypersonic Flow," 23rd International Symposium of Rarefied Gas Dynamics, American Institute of Physics, 2003.

Chapra, S. C., Canale, R. P., Numerical Methods for Engineers, 4th ed., 2002, McGraw Hill, pp. 456-59.

Coleman, G., Stollery, J., "Heat Transfer from Hypersonic Turbulent Flow at a Wedge Compression Corner," Journal of Fluid Mechanics¸ Vol. 56, Part 4, 1972, pp. 741-752.

Gnoffo, P., "CFD Validation Studies for Hypersonic Flow Prediction," AIAA Journal, AIAA 2001-1025, 2001.

Harvey, J. K., "A Review of a Validation Exercise on the Use of the DSMC Method to Compute Viscous/Inviscid Interactions in Hypersonic Flow" 36th AIAA Thermodynamics Conference, AIAA 2003-3643, June 2003.

Harvey, J. K., Holden, M. S., Wadhams, T. P., "Code Validation Study of Laminar Shock/Boundary Layer and Shock/Shock Interactions in Hypersonic Flow," 39th AIAA Aerospace Sciences Meeting, AIAA-2001-1031, AIAA, Reno, NV, January 8-12, 2001.

Hollis, B. R., Horvath, T. J., Berry, S. A., Hamilton II, H., Alter, S. J., "X-33 Computational Aeroheating Predictions and Comparisons with Experimental Data," 33rd AIAA Thermophysics Conference, AIAA-99-3559, AIAA, Norfolk, VA, June 28-July 1, 1999.

Hu, P., Xue, L., Qu, K., Ni, K., "Unified Solver for Modeling and Simulation of Nonlinear Aeroelasticity and Fluid-Structure Interactions," AIAA Atmospheric Flight Mechanics Conference, AIAA 2009-6148, AIAA, Chicago, Illinois, August 10-13, 2009.

Hu, P., Kamakoti, R., Xue L., Wang, Z., Li, Q., Attar, P., Vedula, P., "A Meshless Method for Aeroelastic Applications in ASTE-P Toolset," AIAA Modeling and Simulation Technologies Conference, AIAA-2010-7604, AIAA, Toronto, Ontario, August 2-5, 2010.

Hu, P., Xue L., Kamakoti, R., Zhao, H., Li, Q., Brenner, M. J., "Material Point Method with Least Squares Technique for Nonlinear Aeroelasticity and Fluid-Structure Interactions (FSI) in ASTE-P Toolset," AIAA

Atmospheric Flight Mechanics Conference, AIAA-2010-8224, AIAA, Toronto, Ontario, August 2-5, 2010.

Incropera, F., DeWitt, D., Bergman, T., Lavine, A., Fundamentals of Heat and Mass Transfer, 6th ed., 2007, John Wiley and Sons, pp. 58-60.

Koh, E. P. C., Tsai, H. M., Liu, F., “Euler Solution Using Cartesian Grid with Least Squares Technique,” 41st Aerospace Sciences Meeting and Exhibit, AIAA-2003-1120, AIAA, Reno, NV, January 6-9, 2003.

Moss, J. N., Bird, G. A., Markelov, G. N., “DSMC Simulations of Hypersonic Flows and Comparison with Experiments,” 24th International Symposium of Rarefied Gas Dynamics, American Institute of Physics, 2005.

Moss, J. N., LeBeau, G. J., Glass, C. E., “Hypersonic Shock Interactions About a 25°/65° Sharp Double Cone,” NASA/TM-2002-211778, August 2002.

Moss, J. N., Rault, D. F. G., Price, J. M., “Direct Monte Carlo Simulations of Hypersonic Viscous Interactions Including Separation,” NASA/TM-111528, 1993.

Nompelis, I., “Computational Study of Hypersonic Double-Cone Experiments for Code Validation,” Ph.D. Dissertation, University of Minnesota, 2004.

Panton, R. L., Incompressible Flow, 3rd ed., 2005, John Wiley and Sons, pp. 678-680.

York II, A., Sulsky, D., Schreyer, H., “Fluid-Membrane Interaction Based on the Material Point Method,” International Journal for Numerical Methods in Engineering, Vol. 48, 2000, pp. 901-924.

Wikipedia, Material Point Method, Retrieved December 22, 2012 from http://en.wikipedia.org/wiki/Material_Point_Method